

AD-A037 233

CORNELL UNIV ITHACA N Y SCHOOL OF OPERATIONS RESEARC--ETC F/G 9/2
OPTIMAL POLICY FOR DATABASE BATCH OPERATIONS: BACKUP, CHECKPOIN--ETC(U)
SEP 76 J A MUCKSTADT, G M LOHMAN N00014-75-C-1172

UNCLASSIFIED

TR-312

NL

1 OF 1
AD
A037233



END

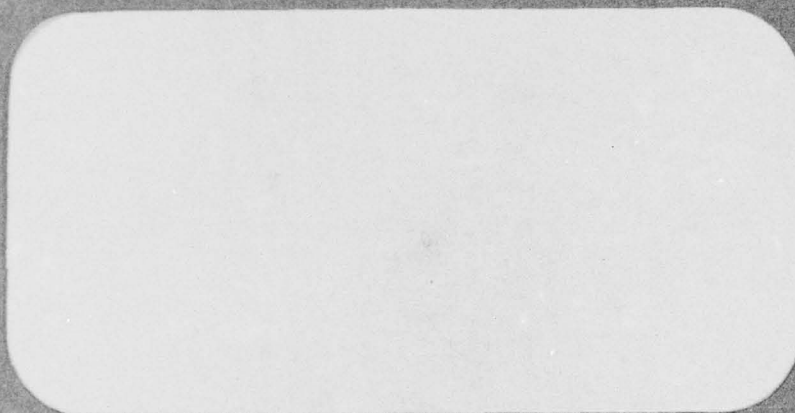
DATE
FILMED
4-77

ADA 037233

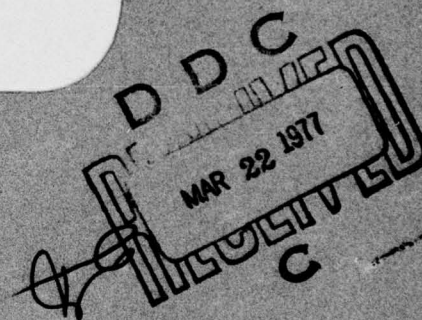
DEPARTMENT
OF
OPERATIONS RESEARCH

Cnw

See 1473

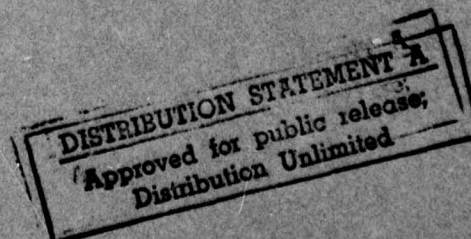


COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION



COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK 14853

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION



SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK

TECHNICAL REPORT NO. 312

September 1976

OPTIMAL POLICY FOR DATABASE BATCH OPERATIONS:
BACKUP, CHECKPOINTING, AND BATCH UPDATE

by

John A. Muckstadt
Guy M. Lohman



This research was supported in part by the Office of Naval Research
under contract N00014-75-C-1172, Task NR 042-335.

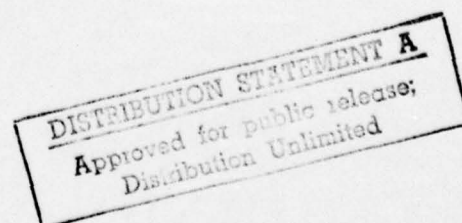


TABLE OF CONTENTS

	Page
SUMMARY	i
Section	
0. INTRODUCTION	1
1. THE OPTIMAL BACKUP POLICY	2
1.1 The Problem - A Brief Tutorial	2
1.2 The Approach	3
1.3 Formulation	4
1.4 Analysis	8
1.5 A Special Case	12
1.6 An Example	13
2. THE OPTIMAL CHECKPOINT POLICY	13
3. THE OPTIMAL BATCH SIZE POLICY	15
4. PRACTICAL CONSIDERATIONS AND EXTENSIONS	16
5. CONCLUSION	17
BIBLIOGRAPHY	18

ACCESSION for ☒ RITE Section ☐ BMT Section

RTS
DDG

UNAPPROVED
JUSTIFICATION

BY

DATE FILE NO. SPECIAL

A

SUMMARY

The purpose of this paper is to present a general model for determining the optimal frequency of batch operations. Specifically, optimal backup, checkpointing, and batch updating policies are derived. Our approach exploits inventory parallels, by seeking the optimal number of items--rather than a time interval--to trigger a batch. The Renewal Reward Theorem is used to find the average long run costs for backup, recovery, and item storage, per unit time, which is then minimized to find the optimal backup policy. This approach allows us to make far less restrictive assumptions about the update arrival process than did previous models, as well as to include storage costs for the updates. The optimal checkpointing and batch updating policies are shown to be special cases of this optimal backup policy. The derivation of previous results as special cases of this model, and an example, demonstrate the generality of the methodology we develop.

0. Introduction.

A common decision database managers must make is how often to perform a batch operation, in order to balance its fixed costs against the variable costs of accumulating items for the next batch. Three variations of this decision are the familiar problems of how often to perform:

(1) a backup of a file in secondary storage, to preserve any real-time updating progress to date on a backup copy of the file for the eventuality of a file loss;

(2) a checkpoint save of main memory, to preserve progress of a long job for the eventuality of a loss of main memory; and

(3) a batch update or reorganization of a file to make it current and/or more compact.

In each problem, more frequent batch operations decrease the variable costs associated with item accumulation, but themselves consume system time and resources. Conversely, batch operations performed less frequently incur the fixed costs less often, but a larger and more costly inventory of items accumulates in the interims between batch operations.

Previous analyses of batching policies have usually considered individually either problem (1) [1,3] or problem (2) [11], although Sayani has linked the two problems with a somewhat generalized model [10]. All of these authors have sought an optimal time interval between batch operations, and all have found equivalent "square root laws" for this interval [6]. The objective is always to minimize the cost of backup and recovery (or saves and reprocessing, in problem (2)), in terms of the processing time or proportion of this time lost; equivalently, system availability may be maximized. Costs are calculated for one cycle of a regenerating process. This cycle may be either one controllable batch cycle [1,3] or one uncontrollable inter-failure period [10,11]. Items are always assumed to arrive at a constant rate, i.e. constant inter-arrival times. For problem (2) this assumption is seen later to be realistic; for

problem (1) it is not. Failures are assumed to occur either at a constant rate [3], or as a Poisson process, i.e. exponentially distributed inter-failure times [1,10,11]. When exponential inter-failure times are assumed, either the Poisson axioms are appealed to [1] or else the resulting exponential terms in the expected cost must be approximated by a truncated Taylor series to obtain a closed-form solution. Usually by assuming that losses are incurred in negligible time, item and failure arrivals are prohibited during backup and recovery (saves and reprocessing). A relaxation of this assumption is shown by Chandy et al. to preclude a closed-form solution for the optimal interval between batches.

In this paper, we describe a generalized model of problem (1). Inventory theory analogies and renewal theorems enable a more precise and general analysis. An optimal backup policy is found via classical optimization techniques. Then we show that previous models, and problems (2) and (3), are special cases of the model we develop. A realistic example is presented to confirm the significance of our generalizations. Finally, extensions and practical considerations are discussed.

1. The Optimal Backup Policy.

1.1 The Problem--A Brief Tutorial

In spite of improved hardware and software reliability, it is inevitable that major failures of a database system will erroneously alter, destroy, and/or leave inconsistent the online files of a database. Backup copies of files must be retained to recover from these file losses, the only failures of concern in problem (1). Since the primary copy of a file is updated in real time, backup copies must be revised periodically, usually by copying the

primary copy en masse to the backup copy. This revision of a redundant copy is called a backup, or dumping. An After-image Log retains, in chronological order, images--called after-images--of all records updated between backups. In the event of a failure, the latest backup is first reloaded to the location of the destroyed primary copy. This restored but outdated copy is then brought up-to-date by restoring--from the After-image Log--the images of all records updated since the backup was made. The latter process, called roll forward, must be done record-by-record. It therefore can be very time-consuming, depending upon how many updates have been completed since the last backup. Updates to the file must be prohibited during both backup and recovery. Frequent backups shorten any necessary roll forward, but themselves consume time and resources [6,8,12].

1.2 The Approach.

The model developed in this section for the generalized backup policy problem is similar to certain continuous review inventory models described in reference 4. This similarity has also been observed by Sayani [10] and Chandy et al. [1]. Furthermore, as pointed out earlier, the optimal time between batch operations was found by all earlier models to follow a square root law similar to the well known "square root formula" from inventory theory. Establishing a fixed time between backup operations in continuous review systems, however, requires stringent assumptions concerning the arrival process for update transactions. If the time between arrivals of updates is described by a random variable, the backup cycle length--the time between completing backup operations--is not constant. Hence in the continuous review case there is no "optimal backup interval." Instead of attempting to find the cycle length, another way of viewing the problem suggested by the continuous

review inventory theory parallel is to find the optimal number of updates to accumulate between commencing backup operations. This is the approach we have taken, because it permits a more thorough analysis requiring fewer assumptions to be made than required by previous authors.

Our analysis also depends heavily on some important results from renewal theory, specifically the Renewal Reward Theorem. Roughly speaking, this theorem states that, under certain mild restrictions, the long run average cost per unit time is equal to the expected cost incurred per cycle divided by the expected cycle length [9]. Thus finding the policy that minimizes this ratio yields the optimal policy for the problem.

By combining the renewal theory and inventory theory concepts, we obtain a simple method for computing the optimal policy. Furthermore, as we shall see later, it was no coincidence that the previous work resulted in the square root formula, because the results are a consequence of the Renewal Reward Theorem.

1.3 Formulation.

Before developing the details of the model, it is helpful to specify some nomenclature and assumptions. The decision variable of the model is n , the number of update transactions per backup cycle. The file has certain fixed parameters that affect the costs and time of each cycle. Let

- b represent the cost[†] incurred to accomplish a backup;
- r represent the cost[†] to reload the file following a failure;
- a represent the cost[†] to restore one record's after-image during roll forward;

[†]Note that these costs should include the opportunity cost of the users' time, as well as hardware costs, since the users are unable to make updates during backup and recovery [6]. See assumption (4) below.

- s represent the cost to store one after-image per unit time;
 β represent the time required to perform a file backup operations
 ρ represent the time required to reload the file following a failure;
 α represent the time required to restore one record's after-image during roll forward.

Let j index the backup cycles ($j = 1, 2, 3, \dots$), and i index update transaction arrivals within a given cycle ($i = 1, 2, \dots, n$). Then define the random variables

- $N(t)$, the number of update transactions at time t that have accumulated since the previous backup operation ($N(t) = 0, 1, \dots, n$);
 X_{ij} , the time between the $(i-1)^{\text{st}}$ and i^{th} transaction arrival in backup cycle j ;
 Y_j , the length of backup cycle j , including backup and recovery times.

Figure 1 displays a graph of $N(t)$ through two backup cycles, given that $n = 5$. The graph also displays the effect of one failure on the length of a cycle, as well as the associated costs. Each U denotes another arrival of an update transaction.

The model is based upon the following set of assumptions. These assumptions, although numerous, are quite reasonable in practice. Furthermore, they are considerably less restrictive than those made in all previous models.

- (1) The random variables X_{ij} and X_{jh} are mutually independent for $i \neq h$ and $j \neq h$. Furthermore, X_{ij} and X_{ih} are identically distributed for each i and $j \neq h$. Consequently, $\sum_{i=1}^n X_{ij}$ and $\sum_{i=1}^n X_{ih}$ are independent and identically distributed random variables for all $j \neq h$. Since the inter-arrival time distributions are the same in each cycle, there is no

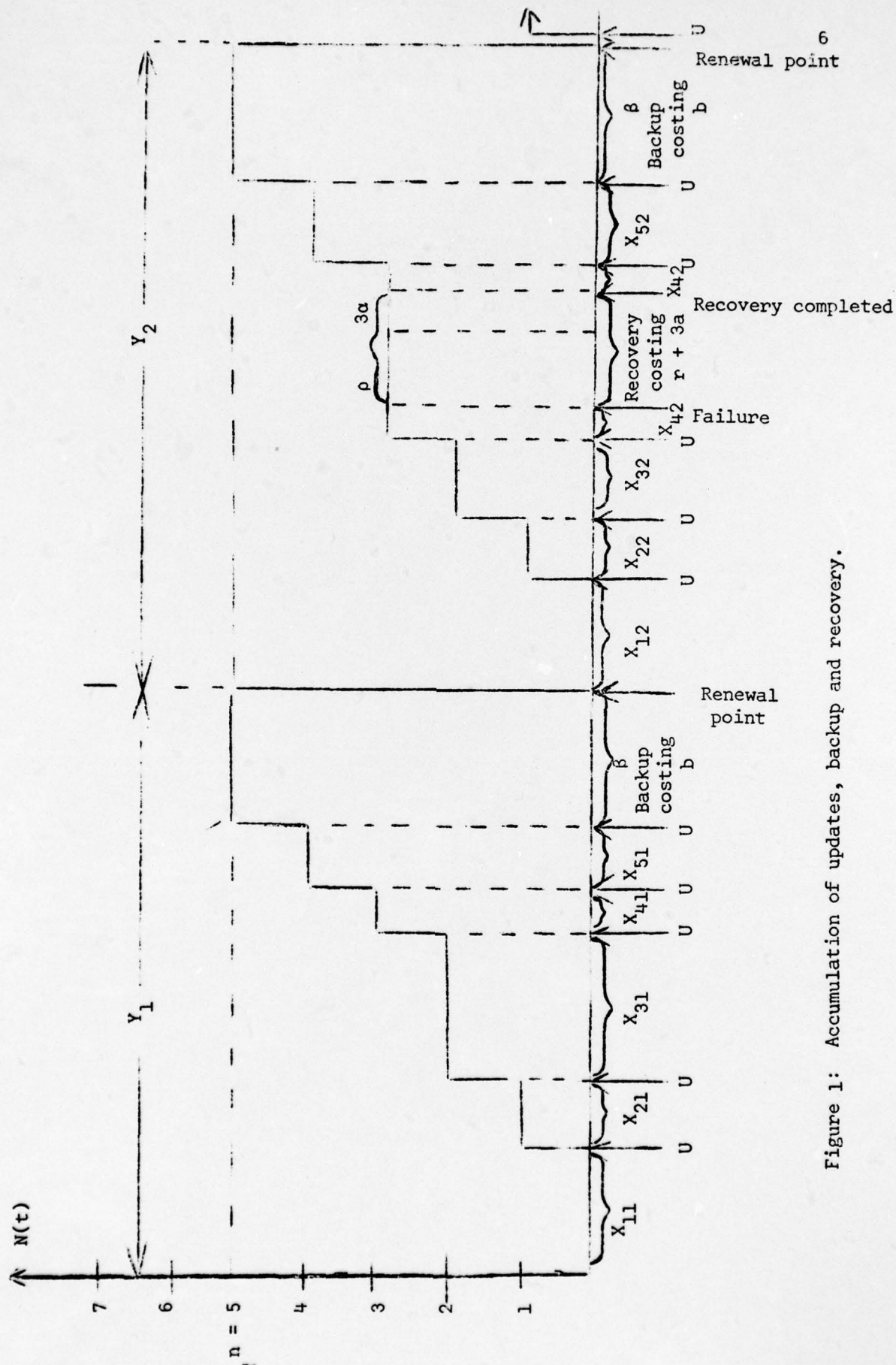


Figure 1: Accumulation of updates, backup and recovery.

need to distinguish one cycle from another, and the cycle subscript, j , on X_{ij} can be dropped.

(2) Failures occur according to a homogeneous Poisson process having rate λ .

(3) The arrival and failure processes are mutually independent random variables.

(4) Neither updates nor failures arrive during the time a backup takes place (β) or a recovery occurs ($\rho + \alpha \cdot N(t)$).

(5) The system is assumed to operate over an infinite period of time.

(6) The time for processing update transactions when they first arrive is negligible compared to the I/O-bound times β , ρ , and α . The cost of such processing is constant with respect to the choice of n , and hence may be ignored.

(7) The distribution of an update inter-arrival period X_{ij} is unaffected by a recovery period that may split X_{ij} (see, for example, X_{42} in Figure 1). When the X_{ij} are exponentially distributed, this assumption is unnecessary.

(8) Each update transaction makes only one update to the file. (Although this assumption is assumed by all earlier models, and can be reasonably justified, omitting it does not alter the approach we have taken for finding the optimal solution.)

Combining assumptions (1) through (5) above, we see that the Y_j are a sequence of independent and identically distributed random variables. Hence the Y_j form a renewal process to which the Renewal Reward Theorem may be applied, and the identical probabilistic characteristics of each backup cycle allow us to consider a typical cycle without the cycle index j . Since all

cycles are alike from a probabilistic viewpoint, we need only consider a typical one. Thus for our analysis of this typical cycle, we will only need the random variables

X_i , the time between the $(i-1)^{\text{st}}$ and i^{th} transaction arrival;

$T = \sum_{i=1}^n X_i$, the length of time in the cycle due to update transaction inter-arrival times;

K_i , the number of failures occurring during X_i ;

M , the number of failures occurring during T ;

R , the length of time spent on recovery in the cycle;

and the expected values that are a function of n ,

$c(n)$, the expected cost of the cycle;

$\tau(n)$, the expected length of the cycle.

The objective is to select the value of n that minimizes the expected long run average cost per unit time. The costs considered are the backup, recovery, and after-image storage costs. Assuming the expected cost and expected length of each cycle are finite, the Renewal Reward Theorem states that the expected long run average cost per unit time of the renewal process Y_j ($j=1,2,3,\dots$) is, with probability 1, equal to $c(n)/\tau(n)$. Thus if we compute both $c(n)$ and $\tau(n)$, we can find the optimal value of n .

1.4 Analysis.

We begin by computing the expected cycle length, $\tau(n)$. As seen in Figure 1,

$\tau(n)$ = expected length of time due to update inter-arrival times
plus the expected time to perform one backup
plus the expected recovery time per cycle

$$\begin{aligned}
&= E \left\{ \sum_{i=1}^n X_i \right\} + \beta + E\{R|n\} \\
&= E\{T|n\} + \beta + E\{R|n\}.
\end{aligned}$$

However, we may express the expected recovery time as

$$E\{R|n\} = \sum_{m=0}^{\infty} E\{R|M=m;n\}P\{M=m|n\}.$$

Additionally, we know from assumptions (1), (2), and (3) that

$$P\{M=m|n\} = \int_0^{\infty} \frac{(\lambda t)^m e^{-\lambda t}}{m!} dG_T(t),$$

where $G_T(t)$ represents the cumulative distribution function for T . But

$$\begin{aligned}
E\{R|M=m;n\} &= m\alpha + \alpha(0 \cdot E\{K_1|M=m;n\} + 1 \cdot E\{K_2|M=m;n\} + \dots \\
&\quad \dots + (n-1) \cdot E\{K_n|M=m;n\}).
\end{aligned}$$

Since the failure process is a homogeneous Poisson process, the time at which a particular failure will occur is uniformly distributed over a backup cycle--excluding the "dead time" portion of the cycle due to backup and recovery. Then this implies that the probability that $N(t) = i - 1$ when a failure occurs is $E\{X_i\}/E\{T|n\}$, where $i = 1, \dots, n$. These results can be rigorously proven by first recognizing that the K_1, \dots, K_n are jointly multinomially distributed, with parameters m, p_1, \dots, p_n , given that $\sum_{i=1}^n K_i = m$. Furthermore, Ross [9] shows that $p_i = E\{X_i\}/E\{T|n\}$, assuming $E\{X_i\} < \infty$. Consequently $E\{K_i|M=m;n\} = m \cdot E\{X_i\}/E\{T|n\}$.

Therefore,

$$\begin{aligned}
 E\{R|n\} &= \left(\rho + \alpha \sum_{i=1}^n (i-1) E\{X_i\} / E\{T|n\} \right) \sum_{m=0}^{\infty} m \int_0^{\infty} \frac{(\lambda t)^m e^{-\lambda t}}{m!} dG_T(t) \\
 &= \left(\rho + \alpha \sum_{i=1}^n (i-1) E\{X_i\} / E\{T|n\} \right) \int_0^{\infty} \sum_{m=0}^{\infty} m \frac{(\lambda t)^m e^{-\lambda t}}{m!} dG_T(t) \\
 &= \left(\rho + \alpha \sum_{i=1}^n (i-1) E\{X_i\} / E\{T|n\} \right) \lambda \int_0^{\infty} t dG_T(t) \\
 &= \left(\rho + \alpha \sum_{i=1}^n (i-1) E\{X_i\} / E\{T|n\} \right) \lambda E\{T|n\}.
 \end{aligned}$$

The operations of integration and summation may be interchanged since the integral is over a product of two functions, each continuous in t and bounded uniformly by 1. Observe that $E\{R|n\}$ does not depend on the form of the distribution function $G_T(t)$, but only on the mean of the distribution! Consequently $\tau(n)$ also depends only on $E\{T|n\}$ and not on $G_T(t)$. The importance of this observation is that the X_i can have any distribution and can be dependent on the length of any other inter-update interval X_h in the same cycle.

Similarly, we may show that

$$c(n) = E\{\text{storage cost per cycle}|n\} + b + E\{\text{recovery cost per cycle}|n\}.$$

where

$$\begin{aligned}
 E\{\text{storage cost per cycle}|n\} &= \sum_{i=1}^n s(i-1)E\{X_i\}, \text{ and} \\
 E\{\text{recovery cost per cycle}|n\} &= \left\{ r + a \sum_{i=1}^n (i-1)E\{X_i\} / E\{T|n\} \right\} \lambda E\{T|n\}.
 \end{aligned}$$

These results also depend only on $E\{T|n\}$ and not on the form of $G_T(t)$.

Combining the above results, we may write the ratio $c(n)/\tau(n)$ for any

given distribution of the X_i , $i=1, \dots, n$. The value of n that minimizes $c(n)/\tau(n)$, call it \hat{n} , is the optimal number of update transactions between the beginning of successive backup operations.

An exact, closed-form solution for \hat{n} can be obtained if we assume that the means of the X_i are identical; that is, $E\{X_i\} = 1/\mu$ for $i = 1, \dots, n$. In this case, $E\{T|n\} = n/\mu$, so that

$$E\{R|n\} = \left(\rho + \alpha \frac{n(n-1)}{2n} \right) \frac{\lambda n}{\mu}, \text{ and}$$

$$\tau(n) = \frac{1}{\mu} \left(n + \rho \lambda n + \frac{\alpha}{2} \lambda (n^2 - n) \right) + \beta.$$

Also,

$$c(n) = \frac{1}{\mu} \left(\frac{s(n^2 - n)}{2} + (r + \frac{a}{2}(n-1))n \lambda \right) + b.$$

Then dividing $c(n)$ by $\tau(n)$ gives the expected long run cost per unit time. Specifically,

$$\frac{c(n)}{\tau(n)} = \frac{(s+a\lambda)n^2 + (2r\lambda - s - a\lambda)n + 2\mu b}{\alpha\lambda n^2 + (2\rho\lambda + 2 - \alpha\lambda)n + 2\mu\beta}.$$

We find the optimal value of n by temporarily assuming that n is a continuous variable. First take the derivative of the ratio $c(n)/\tau(n)$ with respect to n . This yields

$$\begin{aligned} \frac{d}{dn} \frac{c(n)}{\tau(n)} &= [\tau(n)]^{-2} \{ [\alpha\lambda n^2 + (2\rho\lambda + 2 - \alpha\lambda)n + 2\mu\beta] \cdot \\ &\quad [2n(s + a\lambda) + (2r\lambda - s - a\lambda)] \\ &\quad - [2n\alpha\lambda + (2\rho\lambda + 2 - \alpha\lambda)] \cdot \\ &\quad [(s + a\lambda)n^2 + (2r\lambda - s - a\lambda)n + 2\mu b] \} \end{aligned}$$

Setting this derivative equal to zero, we see that n satisfies

$$x\hat{n}^2 + (2\mu y)\hat{n} + \mu z = 0, \text{ or}$$

$$(1) \quad \begin{cases} \hat{n} = \frac{1}{x} \left(-\mu y + \sqrt{\mu^2 y^2 - \mu x z} \right), & x \neq 0, \text{ where} \\ x = \lambda \rho s + s + a\lambda^2 \rho + a\lambda - \alpha\lambda^2 r, \\ y = \beta s + a\beta\lambda - \alpha b\lambda, \text{ and} \\ z = 2\beta\lambda r - 2b\lambda\rho - 2b - y. \end{cases}$$

If $x < 0$, the positive root is the unique minimizing value of n , since the concave parabola $x\hat{n}^2 + (2\mu y)\hat{n} + \mu z$ has a positive slope at the smaller value of \hat{n} . When $x > 0$, which is more likely since a/α will most likely be roughly equal to r/ρ , the positive root clearly yields the unique minimizing value of n . If \hat{n} is not an integer, compute $c(n)/\tau(n)$ for $n = [\hat{n}]$ and $n = [\hat{n}] + 1$, where $[\hat{n}]$ represents the greatest integer less than or equal to \hat{n} . The value of n that yields the smaller value for the ratio is the optimal integral value of n .

1.5 A Special Case.

A special case of the above model occurs when $s = \beta = \rho = \alpha = 0$ (or, equivalently, when $\frac{a}{\alpha} = \frac{b}{\beta} = \frac{r}{\rho}$ and $s = 0$). In this situation we are assuming that backup and recovery times are zero and storage costs are insignificant. In essence, this is the problem studied by Chandy et al., Drake & Smith, Sayani, and Young. For this case, $x = a\lambda$, $y = 0$, and $z = -2b$. Letting \hat{n}_0 represent the optimal solution, equations (1) yield

$$\hat{n}_0 = \sqrt{2b\mu / a\lambda}.$$

Since the expected time between update transactions is $1/\mu$, the expected

length of a backup interval under this optimal policy is

$$(2) \quad \hat{n}_0 \cdot \frac{1}{\mu} = \sqrt{2b/a\lambda\mu}.$$

This is the result obtained by all previous authors [6].[†]

1.6 An Example.

To illustrate the significance of our more generalized model, consider a hypothetical but realistic example. Assume that the primary copy of a file is stored on an IBM 3330 disk, and that the logs are maintained on online tape. Suppose further that $\beta = \rho = 1$ hour = 3600 seconds, $b = r = \$1000$, $s = \$3.2 \cdot 10^{-9}$ /update-second (500 byte records @ 20¢/Kbyte-year, for online tape [7]), $a = \$0.01$ /update, $\alpha = .0467$ second/update [5], $\mu = 2$ /second, and $\lambda = 4 \cdot 10^{-7}$ /second (approximately one loss of the file per month). Then $x = 2.1609 \cdot 10^{-8}$, $y = 7.24 \cdot 10^{-6}$, and $z = -2000$, so that $\hat{n} \approx 743190$, or about once every 4.3 days, for an expected unit cost of \$0.00567/second. The earlier models, which ignore storage costs and significant times to perform backup and recovery, calculate $n_0 =$ one million, or about once every 5.79 days. Sayani's model, which allows significant backup time, derives a backup interval of 5.75 days for this example. The difference of over one day in these backup intervals is primarily due to update storage costs, even though s appears to be an insignificant price.

2. The Optimal Checkpoint Policy.

Accidental loss of the contents of main memory, and especially the all-important registers, is even more common than file losses. Even an interruption of processing due to a power failure or the system "hanging up"

[†]The results originally published by Drake and Smith were in error. By correcting that error their result coincides with expression (2).

can cause the loss of the registers' contents. This type of failure is called a minor failure, because less recovery effort is required than for a file loss. Any job currently being processed when a minor failure occurs must be reprocessed from the last checkpoint: either the beginning of the job or a pause in processing to save partial results on secondary storage. In a multiprocessing system, the pause is usually coordinated among the processes currently active, and a complete copy of main memory is saved. The frequency of checkpoints in a single process or the system as a whole must be determined, to balance the costs of saves and reprocessing [8].

The problem of determining the optimal number of instructions to process between checkpoints, n' , is a special case of the problem solved in the previous section, as can be seen by replacing its parameters with the following analogous parameters for checkpoints. The items accumulated in the checkpoint problem are instructions in the jobs that have been processed since the last checkpoint at the constant rate μ' /second. The batch operation is a save -- either of selected intermediate results of the jobs or all of main memory -- costing b' and requiring time β' . Minor failures occur at a rate of λ' . Recovery from a minor failure requires (1) the reloading of the latest save, costing r' and requiring time ρ' , and (2) reprocessing the instructions of all affected jobs completed since the last checkpoint, each instruction requiring time $\alpha' = 1/\mu'$ and cost a' . Storage costs have no meaning here because instructions are stored both before and after processing regardless of the choice of n . Therefore set $s = 0$. Finally, since costs here are directly proportional to time (i.e. machine cycles) lost, $a'/\alpha' = b'/\beta' = r'/\rho' = 1$. Thus in equations (1), $x = \mu'(a')^2\lambda^1$, $y = 0$, and $z = -2b'$. The optimal number of instructions to execute between checkpoints is therefore

$$n' = (a')^{-1} \sqrt{2b'/\lambda'} ,$$

or an expected inter-checkpoint interval of length

$$n'a' = \sqrt{2b'/\lambda'} ,$$

as found by Young [11]. As before, $[n']$ and $[n'] + 1$ should be tried in the objective function to find the optimal integral solution.

3. The Optimal Batch Size Policy.

When batching updates for a scheduled maintenance, the optimal batch size, n^* , of updates must be determined, in order to balance the fixed costs for a complete pass of the file against the costs of storing and delaying incorporation of those updates into the file [2,7]. The basic model may again be applied to this problem. Replace the update storage cost, s , with a parameter s'' that combines the storage cost and a batching delay penalty for each update, per unit time. Like a backup, a scheduled maintenance requires a complete, sequential pass of the file, costing b'' and taking time β'' . Since file recovery costs are not affected by the frequency of scheduled maintenances in batch systems, set $\lambda = 0$. Assuming as before that the expected update interarrival times are all equal to $1/\mu''$, equations (1) yield $x = s''$, $y = \beta''s''$, $z = -2b'' - \beta''s''$, and

$$n^* = -(\beta''\mu'') + \sqrt{(\beta''\mu'')^2 + \beta''\mu'' + \frac{2b''\mu''}{s''}}$$

The same strategy may be used to find the optimal integral solution. Note that when delay and storage costs are insignificant ($s'' \rightarrow 0$), large batches are permissible ($n^* \rightarrow \infty$); when delay and storage costs are prohibitive ($s'' \rightarrow \infty$) and/or maintenance costs are low ($b'' \rightarrow 0$), real-time updating

($n''=1$) is encouraged (since $\lim_{s \rightarrow \infty} n^* \leq 1$ for all $\beta''\mu'' > 0$).

4. Practical Considerations and Extensions.

The workload of a typical computer system is seldom constant. The operations staff will normally use long-term batch operations such as backup and scheduled maintenances to level the system load during periods of light activity, such as the late night hours. Thus, in practice, the above optimal solutions \hat{n} and n^* are used only as guidelines to pick among nights, say, that a particular file should have a backup or maintenance done. Using the expected interval between these operations, it is clearly advantageous to schedule them so that the effort is spread among the inactive periods as evenly as possible to avoid "bottleneck nights". It is possible to formulate rigorously a model that simultaneously determines the values for \hat{n} and n^* for all files. This model would be formulated so that the possibility of a bottleneck occurring would be arbitrarily small. Similar practical considerations apply to choosing checkpoints.

In section 1.3 we assumed (in assumption (8)) that a single update occurs per transaction. When we relax this assumption, the process $N(t)$ can make jumps of size greater than 1. The length of time to process updates must still be assumed to be small compared to the times between transaction updates. Assume the X_i have a common mean $1/\mu$ for $i = 1, 2, \dots, \ell, \ell \leq n$, and the number of updates generated by any transaction is independent of those generated by all other transactions. Then the probability of $N(t)$ being in any state σ ($\sigma = 0, 1, \dots, n$) in the embedded Markov Chain can be written as weighted sums of the probabilities of u updates per transaction, $u = 1, 2, \dots$. The approach described in section 1 is then directly applicable by using these probabilities rather than the $E\{X_i\}/E\{T|n\}$.

5. Conclusion.

This paper has presented a general model for determining the optimal frequency of batch operations. Specifically, optimal backup, checkpointing, and batch updating policies have been derived. Our approach has exploited inventory parallels, by seeking the optimal number of items--rather than a time interval--to trigger a batch. The Renewal Reward Theorem was used to find the average long run costs for backup, recovery, and item storage, per unit time, which was then minimized to find the optimal backup policy. This approach allowed us to make far less restrictive assumptions about the update arrival process than did previous models, as well as to include storage costs for the updates. The optimal checkpointing and batch updating policies were shown to be special cases of this optimal backup policy. The derivation of previous results as special cases of this model, and an example, have demonstrated the generality of the methodology we have developed.

Bibliography

1. Chandy, K.M., Browne, J.C., Dissly, C.W., and Uhrig, W.R. Analytic models for rollback and recovery strategies in data base systems. IEEE Transaction on Software Engineering SE-1, 1 (March 1975), pp. 100-110.
2. Davis, G.B. Management Information Systems: Conceptual Foundations, Structure, and Development, McGraw-Hill, New York, 1974
3. Drake, R.W. and Smith, J.L. Some techniques for file recovery, Australian Computer J. 3,4 (Nov. 1971), pp. 162-170.
4. Hadley, G. and Whitin, T. Analysis of Inventory Systems, Prentice-Hall Englewood Cliffs, NJ, 1963.
5. IBM Corporation, Intro. to IBM Direct-Access Storage Devices and Organization Methods, Student Text (#GC20-1649-8), IBM Corp., White Plains, NY, Feb. 1974.
6. Lohman, G.M. Optimal data storage and organization in computerized information processing systems subject to failures. Unpublished Ph.D. thesis, School of Opns. Res. and Ind. Engr., Cornell Univ., Ithaca, NY, Jan. 1977.
7. Mader, C. and Hagin, R. Information Systems: Technology, Economics, Applications, Science Research Associates, Chicago, 1974.
8. Martin, J. Security, Accuracy, and Privacy in Computer Systems, Prentice-Hall, Englewood Cliffs, NJ, 1973, pp. 249-264.
9. Ross, S.M. Applied Probability Models with Optimization Applications, Holden-Day, San Francisco, 1970, pp. 85-118.
10. Sayani, H.H. A decision model for restart and recovery from errors in information processing systems. Unpublished Ph.D. thesis, Industrial Engr. Dept., Univ. of Mich., 1973.
11. Young, J.W. A first order approximation to the optimum checkpoint interval. Comm. ACM 17, 9 (Sept. 1974), pp. 530-531.
12. Yourdon, E. Design of On-Line Computer Systems, Prentice-Hall, Englewood Cliffs, NJ, 1972, pp. 340-353, 515-542.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 312 ✓	2. GOVT ACCESSION NO.	3. REPORT'S CATALOG NUMBER
4. TITLE (and Subtitle) OPTIMAL POLICY FOR DATABASE BATCH OPERATIONS: BACKUP, CHECKPOINTING, AND BATCH UPDATE.	5. TYPE OF REPORT & PERIOD COVERED Research Report.	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John A. Muckstadt Guy M. Lohman	8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-1172 ✓	9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 042-335
10. PERFORMING ORGANIZATION NAME AND ADDRESS School of Operations Research and Industrial Engineering, Cornell Univ., Ithaca NY 14853	11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (code 436) Dept. of Navy Arlington, VA 22217	12. REPORT DATE September 1976
13. CONTROLLING OFFICE NAME AND ADDRESS (12) 24p. (14) TR-312	14. NUMBER OF PAGES 18	15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Also supported by the RAND Corporation under Project RAND		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Database batch operations, checkpointing, batch update, Renewal Reward Theorem, Inventory Theory.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The purpose of this paper is to present a general model for determining the optimal frequency of batch operations. Specifically, optimal backup, checkpointin, and batch updating policies are derived. Our approach exploits inventory parallels, by seeking the optimal number of items--rather than a time interval--to trigger a batch. The Renewal Reward Theorem is used to find the average long run costs for backup, recovery, and item storage, per unit time, which is then minimized to find the optimal backup policy. This approach allows us to make far less restrictive assumptions about the update arrival		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-614-66011

Unclassified

409869

✓
Unclassified

CONFIDENTIALITY OF THIS INFORMATION (Data Integrity)

process than did previous models, as well as to include storage costs for the updates. The optimal checkpointing and batch updating policies are shown to be special cases of this optimal backup policy. The derivation of previous results as special cases of this model, and an example, demonstrate the generality of the methodology we develop. **R**

Unclassified